Claims

1.-6. (Canceled)

7. (Currently Amended) In a computer system, a method of annotating computer program code stored on a computer-readable medium, wherein the computer program code is operable to cause a computer to perform according to instructions in the computer program code, the method comprising:

inserting one or more code annotations ~~only~~ at one or more annotation targets;

wherein ~~each of~~ the one or more annotation_s_ comprise at least one annotation on a first pointer to a buffer, wherein the at least one annotation comprises a property that indicates a characteristic of the buffer, wherein the property that indicates the characteristic of the buffer takes a size argument, and wherein the size argument comprises a location of a second pointer associated with the buffer. ~~targets is of an annotation target category selected from a group consisting of: global variable, formal parameter of a function, return value of a function, user-defined data type.~~

8.-13. (Canceled)

14. (Currently Amended) The method of claim ~~13~~ 7 wherein the characteristic is a readable extent of the buffer.

15. (Currently Amended) The method of claim ~~13~~ 7 wherein the characteristic is a writable extent of the buffer.

16.-20. (Canceled)

21. (Currently Amended) The method of claim 7 wherein the ~~one or more code annotations~~ at least one annotation include_s_ an annotation prefix.

22.-23. (Canceled)

24. (Currently Amended) In a computer system, a method of annotating computer program code stored on a computer-readable medium, wherein the computer program code is operable to cause a computer to perform according to instructions in the computer program code, the method comprising:

inserting an annotation at a first value having a first value type in the computer program code;

wherein the annotation is comprises a first instance of a keyword, the first instance of the keyword indicating that the first value has satisfies all usability properties sufficient necessary to allow a first function to rely on the first value, wherein other instances of the keyword identical to the first instance are operable to indicate that other values having different respective value types satisfy all usability properties necessary to allow functions to rely on the respective other values, and wherein the usability properties depend on the value type.

25. (Currently Amended) The method of claim 24 wherein the first value is a formal parameter of the first function.

26. (Currently Amended) The method of claim 24 wherein the first value is a return value.

27. (Currently Amended) The method of claim 24 wherein the first value type is selected from a group comprising: scalar, void, pointer, user-defined type, or struct.

28. (Currently Amended) The method of claim 24 wherein the first value is a reference parameter.

29. (Currently Amended) The method of claim 24 wherein the first value is a pointer, wherein an object pointed to by the pointer has one or more readable elements, the one or more readable elements of the object each having usability properties sufficient to allow the first function to rely on the one or more readable elements.

30. (Currently Amended) In a computer system, a method of annotating computer program code stored on a computer-readable medium, wherein the computer program code is operable to cause a computer to perform according to instructions in the computer program code, the method comprising:

inserting an annotation having an argument in the computer program code, wherein the annotation annotates a value having a first <u>declared</u> value type, and wherein <u>with a first set of</u> usability properties of the value are dependent on the first value type;

wherein the annotation <u>overrides the first set of usability properties of the first declared value type and</u> indicates that the value has usability properties that depend on the properties of a second value type denoted by the argument of the annotation.

31. (Original) The method of claim 30 wherein the first value type is a legacy value type.

32. (Currently Amended) The method of claim 30 wherein the first value type is void * <u>and wherein the second value type has a null-termination characteristic.</u>

33. (Currently Amended) The method of claim 30 wherein the first value type is char * <u>and wherein the second value type has a null-termination characteristic.</u>

34. (Original) In a computer system, a method of annotating computer-executable program code stored on a computer-readable medium, the method comprising:

adding an annotation to the computer program code, wherein the annotation describes a characteristic of a buffer; and

including a size parameter with the annotation, wherein the size parameter describes a portion of the buffer to which the characteristic applies, and wherein the size parameter is operable to describe the portion of the buffer using a size specification selected from a group of plural different size specifications.

35. (Original) The method of claim 34 wherein the group of plural different size specifications comprises: byte count, element count, end pointer location, internal pointer location, sentinel position.

36. (Original) The method of claim 34 wherein the annotation indicates the extent to which the buffer is readable.

37. (Original) The method of claim 34 wherein the annotation indicates the extent to which the buffer is writable.

38.-40. (Canceled)

41. (New) The method of claim 7 further comprising using the location of the second pointer associated with the buffer to determine the size of the buffer.

42. (New) The method of claim 7 wherein the second pointer associated with the buffer is an end pointer for the buffer.

43. (New) The method of claim 7 wherein the second pointer associated with the buffer is an internal pointer for the buffer.

44. (New) The method of claim 14 further comprising using the location of the second pointer associated with the buffer to determine the readable extent of the buffer.

45. (New) The method of claim 15 further comprising using the location of the second pointer associated with the buffer to determine the writable extent of the buffer.

46. (New) The method of claim 24 wherein the annotation further comprises an except qualifier.

47. (New) A method of annotating computer program code stored on a computer-readable medium, the method comprising:

inserting at least one code annotation at a pointer to a data structure comprising plural elements;

wherein the at least one code annotation comprises an explicit dereference qualifier, and wherein the explicit dereference qualifier is operable to specify one or more properties of each of the plural elements of the data structure.

48. (New) The method of claim 47 wherein the data structure is a buffer.

49. (New) The method of claim 47 wherein the data structure is a struct and wherein the plural elements are fields of the struct.

50. (New) A method of processing annotated computer program code stored on a computer-readable medium, the method comprising:

reading from the annotated computer program code at least one annotation on a first pointer to a buffer, wherein the at least one annotation comprises a property that indicates a characteristic of the buffer, wherein the property that indicates the characteristic of the buffer takes a size argument, and wherein the size argument comprises a location of a second pointer associated with the buffer;

processing the annotated computer program code based at least in part on the indicated characteristic of the buffer; and

outputting a result of the processing.

51. (New) The method of claim 50 wherein the characteristic is a readable extent of the buffer, and wherein the processing comprises determining whether the annotated computer program code includes a buffer overrun.

52. (New) The method of claim 50 wherein the characteristic is a writable extent of the buffer, and wherein the processing comprises determining whether the annotated computer program code includes a buffer overrun.

53. (New) A method of processing annotated computer program code stored on a computer-readable medium, the method comprising:

reading at least one annotation having an argument from the annotated computer program code, wherein the at least one annotation annotates a value having a first declared value type with a first set of usability properties, and wherein the annotation overrides the first set of usability properties of the first declared value type and indicates a second set of usability properties for the value that depend on the second value type denoted by the argument of the annotation;

processing the annotated computer program code based at least in part on the second set of usability properties for the value; and

outputting a result of the processing.

54. (New) The method of claim 53 wherein the second value type is a null-terminated string type.

55. (New) The method of claim 53 wherein the processing comprises determining whether the value satisfies the second set of usability properties.